3D Mapping, Localisation and Object Retrieval using Low Cost Robotic Platforms: A Robotic Search Engine for the Real-World

3D mapping, localisation and object retrieval using low cost robotic platforms: A robotic search engine for the real-world

Thomas Whelan*, Michael Kaess†, Ross Finman‡, Maurice Fallon‡, Hordur Johannsson‡, John J. Leonard‡ and John McDonald*

*Department of Computer Science,
National University of Ireland Maynooth,
Co. Kildare, Ireland.
Email: Thomas.J.Whelan@nuim.ie,
johnmcd@cs.nuim.ie

†Robotics Institute,
Carnegie Mellon University,
Pittsburgh, PA 15213, USA.
Email: kaess@cmu.edu

‡Computer Science and
Artificial Intelligence Laboratory,
Massachusetts Institute of Technology,
Cambridge, MA 02139, USA.
Email:{rfinman, mfallon,
hordurj, jleonard}@mit.edu

Abstract—**In this paper we present work in progress on the development of a low-cost autonomous robotic platform that integrates multiple state-of-the-art techniques in RGB-D perception to form a system capable of completing a real-world task in an entirely autonomous fashion. The task we set out to complete is determining the location of a preselected object within the physical world. This experiment requires a robotic framework with a number of capabilities including autonomous exploration, dense real-time localisation and mapping, object detection, path planning and motion control.**

I. Introduction

In this paper we present on-going work to combine a number of recent advances in RGB-D-based perception research to develop an autonomous low-cost wheeled robot. The development of the platform is based around the challenge of autonomously locating a preselected object within the robot's surroundings. The underlying motivation for the work is to investigate the challenges encountered in integrating multiple robust RGB-D based perception techniques which we have developed in our previous research [2, 3, 5, 8, 9], into a single robot framework. The resulting framework includes modules for autonomous exploration, dense real-time localisation and mapping, object detection, path planning and motion control.

Our work is related to a number of other recent efforts that seek to develop and exploit an object-based and/or semantic understanding of a mobile robot's environment. Aydemir et al. [1] investigate techniques for active visual search for objects in a robot's environment, exploiting spatial relationships between objects to develop efficient search strategies. Their paper suggests the use of "dense 3D point cloud representation[s] of scenes to guide the search", which is something that is enabled in our work via the Kintinuous framework for dense RGB-D SLAM. Other related recent work includes the work of Salas-Moreno et al. [6], which develops SLAM++ (an object-oriented approach to SLAM), and Herbst et al. [4], which performs automatic discovery of objects via multiple views of a scene. Also related is a large body of recent work by Saxena et al. [7] which develops techniques for a PR2 robot to detect, classify and grasp objects in a variety of different contexts.

Fig. 1: Photograph of the Turtlebot 2 platform used in our experiment. It is a 2-wheeled platform with an RGB-D sensor, controlled by an onboard computer (a standard laptop in this scenario).

II. System Overview

The experimental setup includes a robotic platform, a laptop (to interface onboard the robot) and a workstation computer. The robot in our experiment is the Clearpath Robotics Turtlebot 2 platform, shown in Figure 1. The laptop onboard the turtlebot is equipped with an Intel Core i7-3630QM CPU,

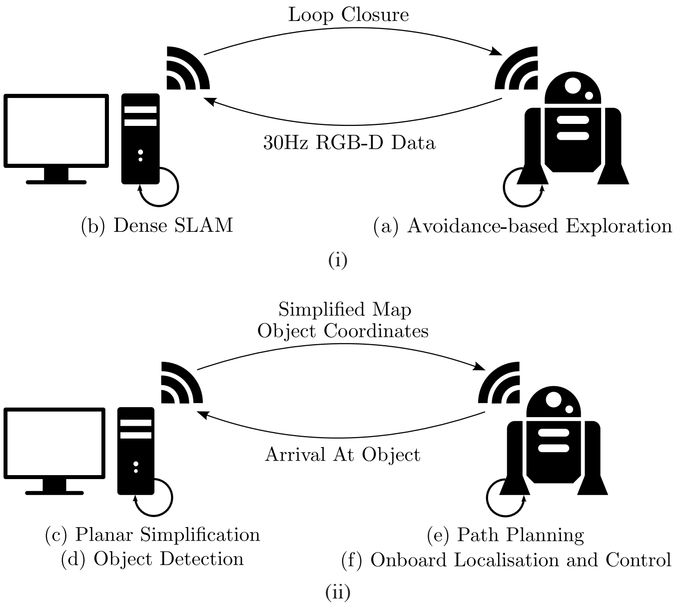


Fig. 2: This figure shows the two main steps involved in executing the required task. The top subfigure (i) illustrates the first step in the process. The robot is set to explore the environment while streaming the RGB-D data captured with the onboard sensor back to the workstation over 802.11n WiFi. The workstation uses this data to reconstruction a globally consistent map of the explored environment in real-time, signalling the robot to cease exploration when a loop has occurred. In the second step, shown in subfigure (ii), the workstation simplifies the dense map such that it is suitable for real-time localisation and detects the position of the desired object in the dense map. This information is sent to the robot which then navigates to the detected object using onboard real-time path planning and control against the simplified map. Details for each of the individual components are provided in the text.

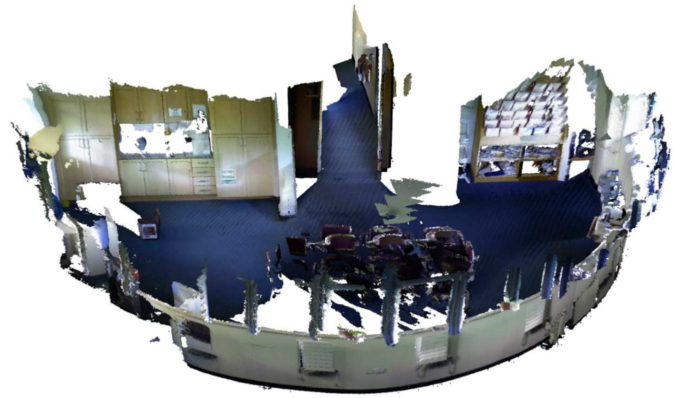


Fig. 3: Dense reconstruction of the area autonomously explored by the robot.

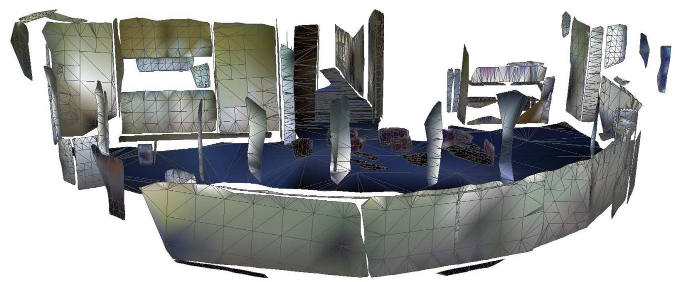


Fig. 4: Planar simplification of the area autonomously explored by the robot with triangulation overlaid.

24GB of RAM and an nVidia GeForce 675M GPU with 2GB of memory. The workstation computer is a standard desktop PC running Ubuntu 12.04 with an Intel Core i7-3960X CPU at 3.30GHz, 16GB of RAM and an nVidia GeForce 680GTX GPU with 2GB of memory. A demonstration of the operation of the complete system can be viewed at:

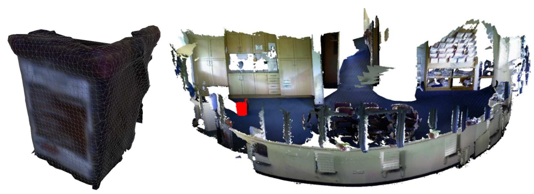
https://www.youtube.com/watch?v=XqDUniEY954.

Figure 2 shows the two main steps involved in the process. Details of each of the steps taken by each of the components labelled alphabetically in Figure 2, are provided below, including the specific instances in the actual experiment carried out.

(a) Avoidance-based Exploration - A simple approach to exploration is adopted in this experiment. Planning is carried out onboard in real-time on the robot itself using the immediate RGB-D data captured with the onboard sensor. The approach attempts to maintain a constant distance to any surfaces to the left of the robot not lying on the ground plane. This is accomplished by analysing a single line of the depth map at the height of the sensor off of the ground plane. The bearing and forward velocity of the robot is recalculated every frame (i.e. at 30Hz) depending on the relative distance to any surfaces detected in the depth map which the robot may collide with. While simple in practice, this method for exploration works well in most environments and is suitable for use as a simple proof-of-concept example. A more sophisticated coverage-based approach could be adopted to improve performance in more complicated environments.

(b) Dense SLAM - In order to reconstruct the environment explored by the robot in real-time, the SLAM approach outlined in [9] is employed on the workstation computer. While the robot explores the real world, the RGB-D data captured by the onboard sensor is streamed in real-time over 802.11n WiFi to the workstation machine. Once a loop closure is detected in the explored area, the robot is signalled to stop exploring. Figure 3 shows the resulting reconstruction of the explored area.

(c) Planar Simplification - As we discuss in Step (f), a simplified planar map of the environment is required for real-time localisation. For this we apply the planar simplification technique described in [5] to the dense reconstruction obtained from Step (b). This model is



(i) (ii)

Fig. 5: From left to right; (i) Learned model of a trash can from a previous scan, with triangulation shown; (ii) Detected position of the object within the mapped environment highlighted in red.

quick to compute and also provides a format in which it is trivial to perform floor plane detection and alignment for 2D path planning (by taking the plane with the largest area with the correct relative transformation to the capturing sensor). Figure 4 shows the simplified planar model computed from the dense map. This compact scene model is transferred to the robot wirelessly for autonomous navigation and localisation in the subsequent steps.

(d) Object Detection - In order to query the robot to navigate to a point of interest, we choose to learn a number of object models as a precursor step the experiment. From this point, provided steps (b) through (c) have succeeded, we can query the system with a known object model. If the system can locate the object within the dense map provided by Step (b), a path can be planned from the last known location of the robot through the environment using the simplified model provided by Step (c). To learn the segmentation parameters for different object models, we use the approach presented by Finman *et al*. [3]. Figure 5 shows a sample object model learned by the system, and the highlighted detection of the object within the dense map provided by Step (b).

(e) Path Planning - The simplified planar model provided by Step (c) includes detection of and alignment with the floor plane. By projecting the remainder of the model onto the floor plane a simple occupancy grid map of the environment can be recovered. From here, the configuration space of the robot can be computed and 2D path planning within the occupancy grid can easily be carried out. In our implementation we seed the path planner with the last known location of the robot and the location of the detected object and run the A* search algorithm to find a path through the occupancy grid. From here we simplify the A* path using a greedy ray-tracing method to get a set of sparse waypoints within the environment. Figure 6 shows the path planned through the environment in our experiment.

(f) Onboard Localisation and Control - Given a simplified model of the environment to localise against and a target point to reach in the map, the robot must autonomously navigate to each point in the planned path in a closed-loop fashion. For this we use the Kinect Monte Carlo Localisation (KMCL) system of Fallon *et al*. [2]. The KMCL system is a particle filter-based localisation system that uses predicted RGB-D frames from within a planar model of an environment as the basis for a likelihood function in comparison to actual RGB-D sensor readings. Figure 7 shows a screenshot of the KMCL system in action during our experiment. The estimated position of the robot is updated at camera frame rate while a simple proportional controller firstly aligns the orientation of the robot with the location of the next waypoint, before adjusting the forward velocity of the robot to reach the waypoint. Once the robot has reached the position of the desired object in the environment, motion is ceased.

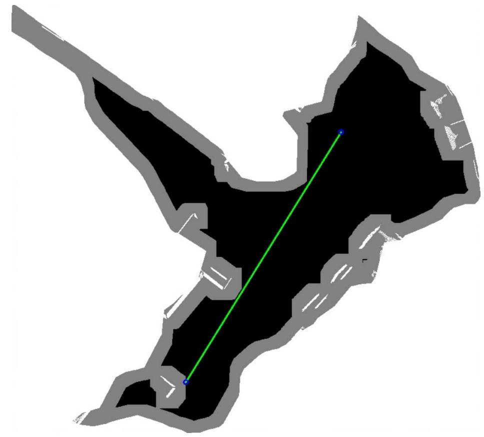


Fig. 6: This figure shows the path planned from the last known location of the robot (top right) to the location of the detected object in the environment. White space is not considered, while grey space is unoccupied but outside of the configuration space of the robot. The path is shown in green while the control waypoints are shown in blue.

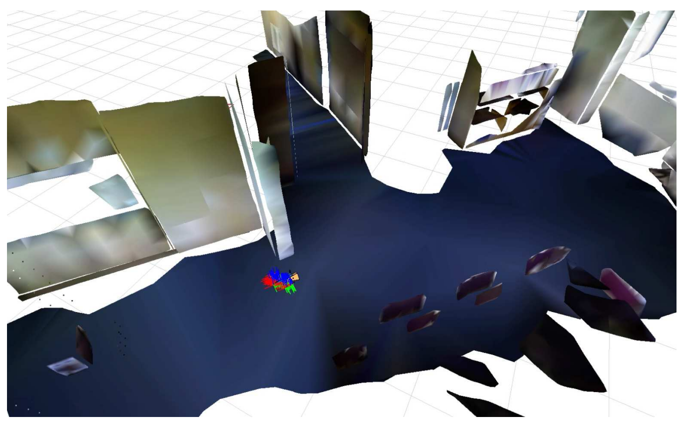


Fig. 7: This figure shows the KMCL system in action as the robot navigates to a target waypoint in the environment in real-time. Shown is the simplified planar model, as well as a number of particle filter estimates of the robot's current position.

III. Discussion and Future Work

The purpose of this case study was to merge a number of recent advances in RGB-D-based perception research to accomplish a simple real-world task using low-cost commodity components. In this sense, the experiment was a success demonstrating clear fitness for purpose. This experiment also highlights the importance of each component of the system and how each is necessary in completing the task. Namely, real-time dense mapping is required to inform the robot that it no longer needs to explore and can immediately access a globally consistent model which can be used for subsequent object detection and future motion planning. The density of the initial map is necessary for performing the detection of a variety of objects commonly found in real-world environments. In contrast to this, quick access to a simplified planar representation of the environment is needed to perform real-time onboard localisation in the mapped area for path planning and motion control.

One observation made during the execution of this experiment was both the compounding of failure rates, and the potential for a cascading effect due to a non-terminal error in the upstream processing resulting in a failure in the downstream processing. As a consequence a number of consecutive runs were required for the robot to complete the task due to the individual rates of failure of each component of the system compounding together. Examples of the individual failures included frame-drops in the wireless streaming of the raw RGB-D image sequence to the SLAM server, failure of the planar segmentation algorithm to robustly estimate the ground plane (*e.g.* due to the errors in the reconstruction process), and failure of the object segmentation and hence recognition due to spurious geometry introduced from noise in the reconstruction process. Of the above, we found that the principal source of error in the system was the unreliable nature of the wireless streaming. This would result in dropped frames which in turn would result in a degradation in camera tracking and 3D scene estimation.

This is one of the key observations of the experiment which highlights the fact that the development of techniques which although alone are quite robust and reliable is not necessarily enough when it comes to combining these techniques together into a complete framework when attempting to accomplish a larger, higher level task.

In future work, we aim to both extend the system's robustness and to quantitatively evaluate its performance and failure modes. In terms of object detection and recognition, we seek to increase the number of objects in the repertoire of the system, drawing on techniques for object-oriented 3D SLAM [6]. We will also explore the challenge of maintaining object-oriented maps in dynamic environments, investigating techniques for efficient "web crawling" for the physical world. Our ultimate aim is to further develop the platform to permit robust long-term autonomous operation over considerably larger scale environments with a much richer semantic searching capability.

Acknowledgments

Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the Irish National Development Plan, the Embark Initiative of the Irish Research Council, by ONR grants N00014-10-1-0936, N00014-11-1-0688, N00014-12-1-0093, and N00014-12-10020, and by NSF award IIS-1318392.

References

[1] A. Aydemir, K. Sjoo, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2818–2824, May 2011. doi: 10.1109/ICRA.2011.5980495.

[2] M. F. Fallon, H. Johannsson, and J. J. Leonard. Efficient scene simulation for robust Monte Carlo localization using an RGB-D camera. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2012.

[3] R. Finman, T. Whelan, M. Kaess, and J.J. Leonard. Toward lifelong object segmentation from change detection in dense RGB-D maps. In *European Conference on Mobile Robots (ECMR)*, Barcelona, Spain, September 2013.

[4] E. Herbst, X. Ren, and D. Fox. RGB-D object discovery via multi-scene analysis. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4850–4856. IEEE, 2011.

[5] L. Ma, T. Whelan, E. Bondarev, P. H. N. de With, and J.B. McDonald. Planar simplification and texturing of dense point cloud maps. In *European Conference on Mobile Robots (ECMR)*, Barcelona, Spain, September 2013.

[6] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Computer Vision and Pattern Recognition (CVPR)*, Portland, Oregon, June 2013.

[7] A. Saxena, L. Wong, M. Quigley, and A. Ng. A vision-based system for grasping novel objects in cluttered environments. In Makoto Kaneko and Yoshihiko Nakamura, editors, *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 337–348. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-14742-5. doi: 10.1007/978-3-642-14743-2_28.

[8] T. Whelan, M. Kaess, M.F. Fallon, H. Johannsson, J.J. Leonard, and J.B. McDonald. Kintinuous: Spatially extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, July 2012.

[9] T. Whelan, M. Kaess, J.J. Leonard, and J.B. McDonald. Deformation-based loop closure for large scale dense RGB-D SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013.